

Команди - типове и структура

ас. Стоян Мечев

катедра „Информационни технологии“
ВВМУ „Н. Й. Вапцаров“

Команди - структура

- Linux е case sensitive, включително за опциите
- Структура: **command** *options* **arguments**
 - **command** какво да правя?
 - **options** как да го направя?
 - -o (кратки опции)
 - --long-option (опции-от-цели-думи)
 - **arguments** върху какво да го приложа?

Още за командите 01

- Видове команди

- Вътрешни `cd`, `break`, `exes` - вградени в шел;
- Външни - съхраняват се в отделна директория, например `/bin` или `/usr/bin`
 - възможно е потребителите да разработят собствени команди;
- `type <command_name>` можете да проверите, дали командата е вътрешна или външна;
 - `type echo`
- `date` - информация за текущата дата.

- Информация за вградени команди

- `help command_name`

- За администратори или за обикновени потребители;

Още за командите 02

- `#` – коментар – от мястото на знака до края на реда;
- няма значение, дали се ползва един интервал или много при разделянето на параметрите;
- Escape symbol `\` – ако желаете да включите специални знаци в командата;
- `\` + нов ред – разделяне на командата на няколко реда.
- при няколко опции `ls -l -a == ls -la`
- Всяка команда завършва с резултат.
 - `0` – командата е успешна
 - `<> 0` – командата не е успешна.
- `echo $?` – показва с каква стойност е приключила предпоследната команда

Специални символи

- | (pipe) изходният поток от една команда става входен поток за друга
- & - фонов режим
- ; разделител на команди – просто изпълнение на командите една след друга
- () субшел – стартира се отделен шел, в който се изпълняват командите в скобите
- < пренасочване на поток от файл
 - `wc < some_file`
- > пренасочване на поток към файл (винаги започва от началото на файла и ако има стари данни те се презаписват)
 - `ls >dir.txt`
- >> добавяне на данни към файл (старите данни се запазват)

Специални символи

Контролни оператори за условно изпълнение на команда

- `||` условно изпълнение на команда при неуспех на предходната команда
 - `command1||command2`
- `&&` условно изпълнение на команда при успех на предходната команда
 - `command1&&command2`

Работа с променливи - `$`

- извличане на стойността променлива
 - `echo $var`
- може да се използва в комбинация с `()`
 - `command1 $(command2)` – `command2` се изпълнява в субшел и резултатът се подава като параметър към `command1`

echo [-n -e -E]

- Работят с опция -e
 - \a Alert (bell)
 - \b Backspace
 - \c Suppress trailing newline (same function as -n option)
 - \f Form feed (clear the screen on a video display)
 - \n New line
 - \r Carriage return
 - \t Horizontal tab
 - \v Vertical tab
 - \uNNNN - униккод символ с шестнадесетичен код NNNN
 - echo -e "\u042F"

Още echo

- `echo line 1;echo line 2; echo line 3`
- `echo line 1&&echo line 2&&echo line 3`
- `echo line 1||echo line 2; echo line 3`
- `(echo In subshell; exit 0) && echo OK || echo Bad exit`
- `(echo In subshell; exit 4) && echo OK || echo Bad exit`

Благодаря за вниманието!
Въпроси?